Assignment #10

Jingye Wang

November 17, 2016

```
library(lme4)
library(foreign)
library(rjags)
library(R2jags)
library(magrittr)
options(stringAsFactors = F)
setwd("~/Dropbox/WUSTL third/Multilevel Modeling for Quantitative Research/assignment/1
0")
.libPaths("/Library/Frameworks/R.framework/Versions/3.3/Resources/library")
radon_data <- load('radon.data')
```

Ch21.1

Uncertainty and variability: for the radon model in Section 21.1, give examples of how the parameter estimates and standard deviations might look if the sample size is increased in the following ways:

Original model

```
radon_model <- function() {
  for (i in 1:n)  {
    y[i] ~ dnorm(y.hat[i], tau.y)
    y.hat[i] <- a[county[i]] + b*x[i]
  }
  b ~ dnorm(0, 0.0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0,100)
  for (j in 1:J)  {a[j] ~ dnorm(mu.a, tau.a)}
  mu.a ~ dnorm(0, 0.0001)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif(0, 100)
}

radon_data_list <- list ("n", "J", "x", "y", "county")
radon_inits <- function() {
  list (a=rnorm(J), b=rnorm(1), mu.a=rnorm(1),
        sigma.y=runif(1), sigma.a=runif(1))
}
radon_param <- c ("a", "b", "mu.a", "sigma.y", "sigma.a")
radon_0 <- jags (radon_data_list, radon_inits, radon_param,
                 model.file = radon_model, n.chains=3, n.iter=2000, DIC = F)
radon_0$BUGSoutput
```

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//RtmpUmdwc
r/modelae510730065.txt", fit using jags,
 3 chains, each with 2000 iterations (first 1000 discarded)
 n.sims = 3000 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     1.2 0.3  0.7  1.0  1.2  1.4   1.7    1  3000
a[2]     0.9 0.1  0.7  0.9  0.9  1.0   1.1    1  3000
a[3]     1.5 0.3  0.9  1.3  1.5  1.7   2.0    1  1800
a[4]     1.5 0.2  1.1  1.4  1.5  1.6   1.9    1  3000
a[5]     1.4 0.2  1.0  1.3  1.4  1.6   1.9    1  3000
a[80]    1.3 0.1  1.1  1.3  1.3  1.4   1.5    1  3000
a[81]    1.9 0.3  1.4  1.7  1.9  2.1   2.5    1  1500
a[82]    1.6 0.3  1.0  1.4  1.6  1.8   2.2    1  2200
a[83]    1.6 0.2  1.2  1.5  1.6  1.7   1.9    1  3000
a[84]    1.6 0.2  1.2  1.5  1.6  1.7   1.9    1  2400
a[85]    1.4 0.3  0.8  1.2  1.4  1.6   2.0    1  1200
b       -0.7 0.1 -0.8 -0.7 -0.7 -0.6  -0.6    1  3000
mu.a     1.5 0.1  1.4  1.4  1.5  1.5   1.6    1  3000
sigma.a  0.3 0.0  0.2  0.3  0.3  0.4   0.4    1   150
sigma.y  0.8 0.0  0.7  0.7  0.8  0.8   0.8    1  3000
```

Part A

4 times as many houses measured within each existing county.

Model 1

```
county_1 <- c()
x_1 <- c()
y_1 <- c()
for (i in unique(county)) {
  county_temp <- county[county == i]
  x_temp <- x[county == i]
  y_temp <- y[county == i]

  county_1 <- c(county_1, rep(county_temp, 4))
  x_1 <- c(x_1, rep(x_temp, 4))
  y_1 <- c(y_1, rep(y_temp, 4))
}
n_1 <- length(x_1)

radon_model_1 <- function() {
  for (i in 1:n_1)  {
    y_1[i] ~ dnorm(y.hat[i], tau.y)
    y.hat[i] <- a[county_1[i]] + b*x_1[i]
  }
  b ~ dnorm(0, 0.0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0,100)
  for (j in 1:J)  {a[j] ~ dnorm(mu.a, tau.a)}
  mu.a ~ dnorm(0, 0.0001)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif(0, 100)
}

radon_data_list_1 <- list ("n_1", "J", "x_1", "y_1", "county_1")
radon_1 <- jags (radon_data_list_1, radon_inits, radon_param,
                 model.file = radon_model_1, n.chains=3, n.iter=2000, DIC = F)
radon_1$BUGSoutput
```

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//RtmpUmdwc
r/modelae56e1eef52.txt", fit using jags,
 3 chains, each with 2000 iterations (first 1000 discarded)
 n.sims = 3000 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     0.9 0.2  0.6  0.8  0.9  1.1   1.3    1  3000
a[2]     0.9 0.0  0.8  0.8  0.9  0.9   1.0    1  2200
a[3]     1.5 0.2  1.1  1.4  1.5  1.6   1.9    1  1800
a[4]     1.5 0.1  1.3  1.5  1.5  1.6   1.8    1  3000
a[5]     1.4 0.2  1.1  1.3  1.4  1.6   1.8    1  3000
a[80]    1.3 0.1  1.2  1.3  1.3  1.4   1.4    1  2500
a[81]    2.5 0.2  2.1  2.3  2.5  2.6   2.9    1  1800
a[82]    1.9 0.3  1.4  1.7  1.9  2.1   2.5    1  2200
a[83]    1.6 0.1  1.4  1.5  1.6  1.7   1.8    1  1900
a[84]    1.6 0.1  1.4  1.6  1.6  1.7   1.8    1  3000
a[85]    1.3 0.2  0.8  1.1  1.3  1.4   1.7    1  3000
b       -0.7 0.0 -0.8 -0.7 -0.7 -0.7  -0.6    1  3000
mu.a     1.5 0.1  1.4  1.5  1.5  1.5   1.6    1  1700
sigma.a  0.4 0.0  0.4  0.4  0.4  0.5   0.5    1  2700
sigma.y  0.7 0.0  0.7  0.7  0.7  0.7   0.7    1  3000
```

This model used 4 times resampling within each existing county. Based on this results, if 4 times as many houses measured within each existing county, then the uncertainty about individual sigma.y and sigma.a declines compared to the original model.

Part B

4 times as many counties, but the same number of houses measured in each county.

Model 2

```
county_2 <- county
for (i in 2:4) {
  county_2 <- c(county_2, county + max(county_2))
}

x_2 <- rep(x, 4)
y_2 <- rep(y, 4)
n_2 <- length(y_2)
J_2 <- J*4

radon_model_2 <- function() {
  for (i in 1:n_2)  {
    y_2[i] ~ dnorm(y.hat[i], tau.y)
    y.hat[i] <- a[county_2[i]] + b*x_2[i]
  }
  b ~ dnorm(0, 0.0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0,100)
  for (j in 1:J_2)  {a[j] ~ dnorm(mu.a, tau.a)}
  mu.a ~ dnorm(0, 0.0001)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif(0, 100)
}

radon_data_list_2 <- list ("n_2", "J_2", "x_2", "y_2", "county_2")
radon_inits_2 <- function() {
  list (a=rnorm(J_2), b=rnorm(1), mu.a=rnorm(1),
        sigma.y=runif(1), sigma.a=runif(1))
}
radon_2 <- jags (radon_data_list_2, radon_inits_2, radon_param,
                 model.file = radon_model_2, n.chains=3, n.iter=2000, DIC = F)
radon_2$BUGSoutput
```

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//RtmpUmdwc
r/modelae530f228f9.txt", fit using jags,
 3 chains, each with 2000 iterations (first 1000 discarded)
 n.sims = 3000 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     1.2 0.2  0.7  1.0  1.2  1.4   1.7    1  3000
a[2]     0.9 0.1  0.7  0.9  0.9  1.0   1.1    1   700
a[3]     1.5 0.3  1.0  1.3  1.5  1.7   2.0    1  3000
a[4]     1.5 0.2  1.1  1.4  1.5  1.7   1.9    1  3000
a[5]     1.4 0.2  0.9  1.3  1.4  1.6   1.9    1  1600
a[335]   1.3 0.1  1.1  1.3  1.3  1.4   1.5    1  3000
a[336]   1.9 0.3  1.4  1.7  1.9  2.1   2.4    1  3000
a[337]   1.6 0.3  1.0  1.4  1.6  1.8   2.2    1  3000
a[338]   1.6 0.2  1.2  1.5  1.6  1.7   1.9    1  3000
a[339]   1.6 0.2  1.2  1.5  1.6  1.7   1.9    1  2100
a[340]   1.4 0.3  0.8  1.2  1.4  1.6   1.9    1  3000
b       -0.7 0.0 -0.8 -0.7 -0.7 -0.7  -0.6    1  3000
mu.a     1.5 0.0  1.4  1.4  1.5  1.5   1.5    1  2400
sigma.a  0.3 0.0  0.3  0.3  0.3  0.3   0.4    1   320
sigma.y  0.8 0.0  0.7  0.7  0.8  0.8   0.8    1  3000
```

If 4 times as many counties, but the same number of houses measured in each county, then the uncertainty about the group-level parameters mu.a and sigma.a decreases compared to the original model.

Part C

4 times as many counties, with 4 times the number of houses measured in each county (thus, 16 times as many houses in total).

Model 3

```
county_3 <- county_1
for (i in 2:4) {
  county_3 <- c(county_3, county_1 + max(county_3))
}
x_3 <- rep(x_1, 4)
y_3 <- rep(y_1, 4)
n_3 <- length(y_3)
J_3 <- unique(county_3) %>% length

radon_model_3 <- function() {
  for (i in 1:n_3)  {
    y_3[i] ~ dnorm(y.hat[i], tau.y)
    y.hat[i] <- a[county_3[i]] + b*x_3[i]
  }
  b ~ dnorm(0, 0.0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0,100)
  for (j in 1:J_3)  {a[j] ~ dnorm(mu.a, tau.a)}
  mu.a ~ dnorm(0, 0.0001)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif(0, 100)
}

radon_data_list_3 <- list ("n_3", "J_3", "x_3", "y_3", "county_3")
radon_inits_3 <- function() {
  list (a=rnorm(J_3), b=rnorm(1), mu.a=rnorm(1),
        sigma.y=runif(1), sigma.a=runif(1))
}
radon_3 <- jags (radon_data_list_3, radon_inits_3, radon_param,
                 model.file = radon_model_3, n.chains=3, n.iter=2000, DIC = F)
radon_3$BUGSoutput
```

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//RtmpUmdwc
r/modelae56a600c89.txt", fit using jags,
 3 chains, each with 2000 iterations (first 1000 discarded)
 n.sims = 3000 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     0.9 0.2  0.6  0.8  0.9  1.1   1.3    1  3000
a[2]     0.9 0.1  0.8  0.8  0.9  0.9   1.0    1  3000
a[3]     1.5 0.2  1.1  1.4  1.5  1.6   1.9    1  2100
a[4]     1.5 0.1  1.3  1.5  1.5  1.6   1.8    1  3000
a[5]     1.4 0.2  1.1  1.3  1.4  1.6   1.8    1  3000
a[335]   1.3 0.1  1.2  1.3  1.3  1.4   1.4    1  3000
a[336]   2.5 0.2  2.1  2.3  2.5  2.6   2.8    1  2600
a[337]   1.9 0.3  1.4  1.7  1.9  2.1   2.5    1  1400
a[338]   1.6 0.1  1.4  1.5  1.6  1.7   1.8    1  3000
a[339]   1.6 0.1  1.4  1.6  1.6  1.7   1.8    1  3000
a[340]   1.3 0.2  0.8  1.1  1.3  1.4   1.7    1  2000
b       -0.7 0.0 -0.7 -0.7 -0.7 -0.7  -0.7    1  1600
mu.a     1.5 0.0  1.4  1.5  1.5  1.5   1.5    1  3000
sigma.a  0.4 0.0  0.4  0.4  0.4  0.4   0.5    1   940
sigma.y  0.7 0.0  0.7  0.7  0.7  0.7   0.7    1  1400
```

If 4 times as many counties, with 4 times the number of houses measured in each county, then both individual αj's and the uncertainty about the group-level parameters mu.a and sigma.a would be expected to decline.

Ch21.3

Superpopulation and finite-population standard deviations

Part A

Fit a varying-intercept, varying-slope model to the data in the folder radon. Get point estimates and 50% intervals for the superpopulation and finite-population standard deviations.

Model 3a

```
radon_model_3a <- function() {
  for (i in 1:n)  {
    y[i] ~ dnorm(y.hat[i], tau.y)
    y.hat[i] <- a[county[i]] + b[county[i]]*x[i]
    e.y[i] <- y[i] - y.hat[i]
  }
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0,100)
  s.y <- sd(e.y[])

  for (j in 1:J)  {
    a[j] ~ dnorm(a.hat[j], tau.a)
    b[j] ~ dnorm(b.hat[j], tau.b)
    a.hat[j] <- mu.a
    b.hat[j] <- mu.b
    e.a[j] <- a[j] - a.hat[j]
    e.b[j] <- b[j] - b.hat[j]
  }
  mu.a ~ dnorm(0, 0.0001)
  mu.b ~ dnorm(0 ,0.0001)
  tau.a <- pow(sigma.a, -2)
  tau.b <- pow(sigma.b, -2)
  sigma.a ~ dunif(0, 100)
  sigma.b ~ dunif(0, 100)
  s.a <- sd(e.a[])
  s.b <- sd(e.b[])
}

radon_data_list_3a <- list ("n", "J", "x", "y", "county")
radon_param_3a <- c ("a", "b", "sigma.a", "sigma.b", "sigma.y", "s.a", "s.b", "s.y")

radon_inits_3a <- function () {
  list (a=rnorm(J), b=rnorm(J), mu.a=rnorm(1), mu.b=rnorm(1),
        sigma.y=runif(1), sigma.a=runif(1), sigma.b=runif(1))
}

radon_3a <- jags (radon_data_list_3a, radon_inits_3a, radon_param_3a,
                 model.file = radon_model_3a, n.chains=3, n.iter=2000, DIC = F)
radon_3a$BUGSoutput
```

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//RtmpUmdwc
r/modelae547d72cee.txt", fit using jags,
 3 chains, each with 2000 iterations (first 1000 discarded)
 n.sims = 3000 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     1.2 0.3  0.7  1.0  1.2  1.4   1.7  1.0   900
a[2]     0.9 0.1  0.7  0.9  0.9  1.0   1.1  1.0  3000
a[3]     1.5 0.3  0.9  1.3  1.5  1.7   2.0  1.0  3000
a[4]     1.5 0.2  1.1  1.4  1.5  1.7   2.0  1.0  3000
a[5]     1.4 0.3  0.9  1.3  1.4  1.6   2.0  1.0  3000
a[80]    1.4 0.1  1.1  1.3  1.4  1.4   1.6  1.0  1400
a[81]    1.9 0.3  1.3  1.7  1.9  2.1   2.5  1.0   550
a[82]    1.6 0.3  1.0  1.4  1.6  1.8   2.2  1.0  2000
a[83]    1.6 0.2  1.2  1.5  1.6  1.7   2.0  1.0   120
a[84]    1.6 0.2  1.2  1.5  1.6  1.7   2.0  1.0  3000
a[85]    1.4 0.3  0.8  1.2  1.4  1.6   2.0  1.0  1700
b[1]    -0.7 0.2 -1.2 -0.8 -0.7 -0.5  -0.1  1.0   290
b[2]    -0.8 0.2 -1.4 -0.9 -0.8 -0.7  -0.4  1.1    72
b[3]    -0.7 0.2 -1.1 -0.8 -0.7 -0.6  -0.1  1.0   240
b[4]    -0.7 0.2 -1.2 -0.8 -0.7 -0.6  -0.3  1.0  3000
b[5]    -0.7 0.2 -1.1 -0.8 -0.7 -0.6  -0.1  1.0   460
b[80]   -0.8 0.2 -1.3 -0.9 -0.8 -0.7  -0.4  1.1    56
b[81]   -0.5 0.3 -1.0 -0.7 -0.6 -0.4   0.1  1.1    32
b[82]   -0.7 0.3 -1.2 -0.8 -0.7 -0.6  -0.1  1.0  3000
b[83]   -0.9 0.3 -1.6 -1.1 -0.8 -0.7  -0.5  1.1    20
b[84]   -0.7 0.2 -1.2 -0.8 -0.7 -0.6  -0.2  1.0  2500
b[85]   -0.7 0.3 -1.3 -0.8 -0.7 -0.6  -0.1  1.0  1100
s.a      0.3 0.0  0.3  0.3  0.3  0.4   0.4  1.0   250
s.b      0.2 0.1  0.0  0.1  0.2  0.3   0.5  1.9     6
s.y      0.8 0.0  0.7  0.7  0.8  0.8   0.8  1.1    27
sigma.a  0.3 0.0  0.3  0.3  0.3  0.4   0.4  1.0   180
sigma.b  0.2 0.1  0.0  0.1  0.2  0.3   0.5  1.9     6
sigma.y  0.8 0.0  0.7  0.7  0.8  0.8   0.8  1.0   560
```

Based on the 50% intervals in the results, the inferences for the superpopulation standard deviations are more uncertain than those for the finite-population deviations.

Part B

Repeat step (a) but just using a sample of 10 counties. The inference for the superpopulation standard deviation should now be much more uncertain than the finite-population standard deviation.

Model 3b

```
set.seed(111)
smp_idx <- county %in% (county %>% unique %>% sample(10))
county_3b_raw <- county[smp_idx]
# convert raw county index to a new index of c(1:10)
county_3b <- county_3b_raw %>% factor %>% as.numeric
x_3b <- x[smp_idx]
y_3b <- y[smp_idx]
n_3b <- length(y_3b)
J_3b <- 10

radon_model_3b <- function() {
  for (i in 1:n_3b)  {
    y_3b[i] ~ dnorm(y.hat[i], tau.y)
    y.hat[i] <- a[county_3b[i]] + b[county_3b[i]]*x_3b[i]
    e.y[i] <- y_3b[i] - y.hat[i]
  }
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0,100)
  s.y <- sd(e.y[])

  for (j in 1:J_3b)  {
    a[j] ~ dnorm(a.hat[j], tau.a)
    b[j] ~ dnorm(b.hat[j], tau.b)
    a.hat[j] <- mu.a
    b.hat[j] <- mu.b
    e.a[j] <- a[j] - a.hat[j]
    e.b[j] <- b[j] - b.hat[j]
  }
  mu.a ~ dnorm(0, 0.0001)
  mu.b ~ dnorm(0 ,0.0001)
  tau.a <- pow(sigma.a, -2)
  tau.b <- pow(sigma.b, -2)
  sigma.a ~ dunif(0, 100)
  sigma.b ~ dunif(0, 100)
  s.a <- sd(e.a[])
  s.b <- sd(e.b[])
}

radon_data_list_3b <- list ("n_3b", "J_3b", "x_3b", "y_3b", "county_3b")
radon_param_3b <- c ("a", "b", 'sigma.a', 'sigma.b', 'sigma.y', "s.a", "s.b", "s.y")

radon_inits_3b <- function () {
  list (a=rnorm(J_3b), b=rnorm(J_3b), mu.a=rnorm(1), mu.b=rnorm(1),
        sigma.y=runif(1), sigma.a=runif(1), sigma.b=runif(1))
}

radon_3b <- jags (radon_data_list_3b, radon_inits_3b, radon_param_3b,
                  model.file = radon_model_3b, n.chains=3, n.iter=2000, DIC = F)
radon_3b$BUGSoutput
```

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//RtmpUmdwc
r/modelae57fc6e21a.txt", fit using jags,
 3 chains, each with 2000 iterations (first 1000 discarded)
 n.sims = 3000 iterations saved
        mean  sd 2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     1.2 0.4  0.5  1.0  1.3  1.5   1.9  1.0  3000
a[2]     1.9 0.3  1.3  1.7  1.9  2.1   2.6  1.0  2200
a[3]     1.9 0.3  1.4  1.7  1.9  2.1   2.4  1.0  2500
a[4]     1.8 0.4  1.1  1.6  1.8  2.0   2.6  1.0  2400
a[5]     1.7 0.4  0.9  1.4  1.7  1.9   2.4  1.0   540
a[6]     2.1 0.3  1.5  1.9  2.1  2.3   2.8  1.0  3000
a[7]     2.0 0.3  1.4  1.8  2.0  2.1   2.6  1.0  3000
a[8]     1.7 0.3  1.2  1.6  1.7  1.9   2.3  1.0  2400
a[9]     1.4 0.1  1.2  1.3  1.4  1.5   1.6  1.0  2200
a[10]    1.8 0.2  1.5  1.7  1.8  2.0   2.3  1.0   620
b[1]    -0.7 0.7 -1.9 -1.2 -0.7 -0.2   0.7  1.0   690
b[2]    -0.8 0.5 -1.9 -1.2 -0.9 -0.5   0.3  1.0   380
b[3]    -1.0 1.4 -3.6 -1.7 -1.0 -0.3   1.9  1.0   580
b[4]    -1.1 0.6 -2.2 -1.4 -1.1 -0.7   0.1  1.0   750
b[5]    -1.0 1.4 -3.7 -1.7 -1.0 -0.3   1.8  1.0   690
b[6]    -1.8 0.5 -2.7 -2.2 -1.8 -1.5  -1.0  1.0  3000
b[7]    -1.0 1.4 -3.9 -1.8 -1.0 -0.3   1.9  1.0  1000
b[8]     0.4 0.9 -1.2 -0.2  0.4  1.0   2.1  1.0   160
b[9]    -1.1 0.3 -1.7 -1.3 -1.1 -0.9  -0.4  1.0   840
b[10]   -1.8 0.5 -2.7 -2.1 -1.8 -1.5  -0.9  1.0  1200
s.a      0.4 0.1  0.1  0.3  0.4  0.4   0.6  1.1   370
s.b      1.0 0.4  0.2  0.7  0.9  1.2   1.9  1.0   140
s.y      0.7 0.0  0.7  0.7  0.7  0.7   0.8  1.0  1800
sigma.a  0.4 0.2  0.1  0.3  0.4  0.5   0.9  1.0   750
sigma.b  1.1 0.6  0.2  0.7  1.0  1.4   2.6  1.0   130
sigma.y  0.7 0.1  0.6  0.7  0.7  0.8   0.9  1.0   850
```

Based on this result, the interferences for the superpopulation standard deviation became more uncertain than those of the finite population standard deviation as compared with the previous question.

Part C

Repeat but just using a sample of 5 counties. The inferences should be even more different now.

Model 3c

```
set.seed(111)
smp_idx <- county %in% (county %>% unique %>% sample(5))
county_3c_raw <- county[smp_idx]
# convert raw county index to a new index of c(5)
county_3c <- county_3c_raw %>% factor %>% as.numeric
x_3c <- x[smp_idx]
y_3c <- y[smp_idx]
n_3c <- length(y_3c)
J_3c <- 5

radon_model_3c <- function() {
  for (i in 1:n_3c)  {
    y_3c[i] ~ dnorm(y.hat[i], tau.y)
    y.hat[i] <- a[county_3c[i]] + b[county_3c[i]]*x_3c[i]
    e.y[i] <- y_3c[i] - y.hat[i]
  }
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0,100)
  s.y <- sd(e.y[])

  for (j in 1:J_3c)  {
    a[j] ~ dnorm(a.hat[j], tau.a)
    b[j] ~ dnorm(b.hat[j], tau.b)
    a.hat[j] <- mu.a
    b.hat[j] <- mu.b
    e.a[j] <- a[j] - a.hat[j]
    e.b[j] <- b[j] - b.hat[j]
  }
  mu.a ~ dnorm(0, 0.0001)
  mu.b ~ dnorm(0 ,0.0001)
  tau.a <- pow(sigma.a, -2)
  tau.b <- pow(sigma.b, -2)
  sigma.a ~ dunif(0, 100)
  sigma.b ~ dunif(0, 100)
  s.a <- sd(e.a[])
  s.b <- sd(e.b[])
}

radon_data_list_3c <- list ("n_3c", "J_3c", "x_3c", "y_3c", "county_3c")
radon_param_3c <- c ("a", "b", 'sigma.a', 'sigma.b', 'sigma.y', "s.a", "s.b", "s.y")

radon_inits_3c <- function () {
  list (a=rnorm(J_3c), b=rnorm(J_3c), mu.a=rnorm(1), mu.b=rnorm(1),
        sigma.y=runif(1), sigma.a=runif(1), sigma.b=runif(1))
}

radon_3c <- jags (radon_data_list_3c, radon_inits_3c, radon_param_3c,
                  model.file = radon_model_3c, n.chains=3, n.iter=2000, DIC = F)
radon_3c$BUGSoutput
```

```
Inference for Bugs model at "/var/folders/ld/tp92rb3n3gs4zx9_3_xly5fw0000gn/T//RtmpUmdwc
r/modelae55a265a7f.txt", fit using jags,
 3 chains, each with 2000 iterations (first 1000 discarded)
 n.sims = 3000 iterations saved
        mean   sd  2.5%  25%  50%  75% 97.5% Rhat n.eff
a[1]     2.0  0.2   1.6  1.9  2.0  2.1   2.5  1.0  3000
a[2]     2.2  0.3   1.8  2.0  2.2  2.4   2.8  1.0   230
a[3]     2.1  0.2   1.6  1.9  2.1  2.2   2.6  1.0   680
a[4]     1.9  0.3   1.2  1.7  1.9  2.0   2.3  1.0   160
a[5]     2.0  0.2   1.6  1.9  2.0  2.1   2.3  1.0  2300
b[1]    -0.2 20.6 -29.0 -3.9 -0.9  2.0  40.4  1.2   430
b[2]    -2.1  0.4  -2.9 -2.3 -2.1 -1.8  -1.3  1.0   650
b[3]    -0.3 21.1 -28.8 -3.7 -1.0  2.0  34.6  1.2   710
b[4]     1.0  0.8  -0.6  0.5  1.0  1.5   2.5  1.0   210
b[5]    -2.1  0.4  -3.0 -2.4 -2.1 -1.8  -1.2  1.0   710
s.a      0.2  0.2   0.0  0.1  0.2  0.3   0.6  1.1    50
s.b      6.0 11.4   0.9  1.7  2.6  5.0  40.9  1.1    38
s.y      0.7  0.0   0.6  0.6  0.7  0.7   0.8  1.0   500
sigma.a  0.3  0.3   0.0  0.1  0.2  0.4   1.2  1.1    47
sigma.b  8.9 15.2   0.9  2.2  3.7  8.1  68.4  1.1    38
sigma.y  0.7  0.1   0.5  0.6  0.7  0.7   0.9  1.0   540
```

Based on this result, the interferences for the superpopulation standard deviation became even more uncertain than those of the finite population standard deviation as compared with the previous question.

Ch21.4

Contrasts: fit in Bugs a varying-intercept model to the radon data with log uranium as a group-level predictor. You will compare inferences for the superpopulation contrast (that is, the slope for log uranium in the county-level model) and the corresponding finite-population contrast (that is, the coefficient of log uranium for the intercepts for the particular counties in the data). You will need to postprocess the simulations in R in order to get simulations for the finite-population contrast.

Part A

Compare the inferences (estimates and standard errors) for the superpopulation and finite-population contrasts.

Model 4a

```
county_4a <- county
x_4a <- x
u_4a <- u

y_4a <- y
n_4a <- length(y_4a)
J_4a <- J

radon_model_4a <- function() {
  for (i in 1:n_4a)  {
    y_4a[i] ~ dnorm(y.hat[i], tau.y)
    y.hat[i] <- a[county_4a[i]] + b*x_4a[i]
  }
  b ~ dnorm(0, 0.0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0,100)

  for (j in 1:J_4a)  {
    a[j] ~ dnorm(a.hat[j], tau.a)
    a.hat[j] <- g.0 + g.1*u_4a[j]
  }
  g.0 ~ dnorm(0, 0.0001)
  g.1 ~ dnorm(0 ,0.0001)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif(0, 100)
}

radon_data_list_4a <- list ("n_4a", "J_4a", "x_4a", "y_4a", "county_4a", 'u_4a')
radon_param_4a <- c ("a", "b", 'sigma.a', 'sigma.y', 'g.0', 'g.1')

radon_inits_4a <- function () {
  list (a=rnorm(J_4a), b=rnorm(1), g.0=rnorm(1), g.1=rnorm(1),
        sigma.y=runif(1), sigma.a=runif(1))
}

radon_4a <- jags (radon_data_list_4a, radon_inits_4a, radon_param_4a,
                  model.file = radon_model_4a, n.chains=3, n.iter=2000, DIC = F)

attach.jags(radon_4a)
finite.g.1 <- rep(NA, n.sims)
for (s in 1:n.sims) {
  finite.pop <- lm(a[s, ] ~ u_4a)
  finite.g.1[s] <- coef(finite.pop)['u_4a']
}

quantile(finite.g.1, c(.025, .975))
mean(finite.g.1)
sd(finite.g.1)
detach()
```

```
quantile(finite.g.1, c(.025, .975))
     2.5%     97.5% 
0.5679942 0.8880594 
```

```
mean(finite.g.1)
[1] 0.7222979
```

```
sd(finite.g.1)
[1] 0.08094942
```

The mean is 0.722, which is a little higher than the superpopulation. The sd is 0.081, which is a little lower than the superpopulation.

Part B

Repeat part (a), but fitting the model just to the first three counties in the dataset.

Model 4b

```
idx_4b <- county %in% c(1:3)
county_4b <- county[idx_4b]
x_4b <- x[idx_4b]
u_4b <- u[unique(county_4b)]
## You need to check if u is log transformed already ?
y_4b <- y[idx_4b]
n_4b <- length(y_4b)
J_4b <- 3

radon_model_4b <- function() {
  for (i in 1:n_4b)  {
    y_4b[i] ~ dnorm(y.hat[i], tau.y)
    y.hat[i] <- a[county_4b[i]] + b*x_4b[i]
  }
  b ~ dnorm(0, 0.0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif(0,100)

  for (j in 1:J_4b)  {
    a[j] ~ dnorm(a.hat[j], tau.a)
    a.hat[j] <- g.0 + g.1*u_4b[j]
  }
  g.0 ~ dnorm(0, 0.0001)
  g.1 ~ dnorm(0 ,0.0001)
  tau.a <- pow(sigma.a, -2)
  sigma.a ~ dunif(0, 100)
}

radon_data_list_4b <- list ("n_4b", "J_4b", "x_4b", "y_4b", "county_4b", 'u_4b')
radon_param_4b <- c ("a", "b", 'sigma.a', 'sigma.y', 'g.0', 'g.1')

radon_inits_4b <- function () {
  list (a=rnorm(J_4b), b=rnorm(1), g.0=rnorm(1), g.1=rnorm(1),
        sigma.y=runif(1), sigma.a=runif(1))
}

radon_4b <- jags (radon_data_list_4b, radon_inits_4b, radon_param_4b,
                  model.file = radon_model_4b, n.chains=3, n.iter=2000, DIC = F)
attach.jags(radon_4b)
finite.g.1 <- rep(NA, n.sims)
for (s in 1:n.sims) {
  finite.pop <- lm(a[s, ] ~ u_4b)
  finite.g.1[s] <- coef(finite.pop)['u_4b']
}

quantile(finite.g.1, c(.025, .975))
mean(finite.g.1)
sd(finite.g.1)
detach()
```

```
quantile(finite.g.1, c(.025, .975))
      2.5%      97.5% 
-0.2909402  2.4425692 
```

```
mean(finite.g.1)
[1] 1.047293
```

```
sd(finite.g.1)
[1] 0.704779
```

The mean is 1.047, which is higher than the superpopulation. The sd is 0.705, which is lower than the superpopulation.

Ch22.1

Take a varying-intercept model from one of the exercises in Part 2 of this book and construct the corresponding ANOVA plot as in Section 22.3.

```
# Preparation
library("arm")
pilots <- read.table ("pilots.dat", header=TRUE)
attach.all (pilots)
group.names <- as.vector(unique(group))
scenario.names <- as.vector(unique(scenario))
n.group <- length(group.names)
n.scenario <- length(scenario.names)
successes <- NULL
failures <- NULL
group.id <- NULL
scenario.id <- NULL
for (j in 1:n.group){
  for (k in 1:n.scenario){
    ok <- group==group.names[j] & scenario==scenario.names[k]
    successes <- c (successes, sum(recovered[ok]==1,na.rm=T))
    failures <- c (failures, sum(recovered[ok]==0,na.rm=T))
    group.id <- c (group.id, j)
    scenario.id <- c (scenario.id, k)
  }
}

y <- successes/(successes+failures)
treatment <- group.id
airport <- scenario.id

## Fit the model using Bugs
pilot_model <- function() {
  for (i in 1:n){
    y[i] ~ dnorm (y.hat[i], tau.y)
    y.hat[i] <- mu + gamma[treatment[i]] + delta[airport[i]]
  }
  mu ~ dnorm (0, .0001)
  tau.y <- pow(sigma.y, -2)
  sigma.y ~ dunif (0, 100)

  for (j in 1:n.treatment){
    gamma[j] ~ dnorm (0, tau.gamma)
  }
  tau.gamma <- pow(sigma.gamma, -2)
  sigma.gamma ~ dunif (0, 100)

  for (k in 1:n.airport){
    delta[k] ~ dnorm (0, tau.delta)
  }
  tau.delta <- pow(sigma.delta, -2)
  sigma.delta ~ dunif (0, 100)
}

n.treatment <- max(treatment)
n.airport <- max(airport)
n <- length(y)
```

```
data <- list ("y", "treatment", "airport", "n", "n.treatment", "n.airport")
inits <- function (){
  list (mu=rnorm(1), sigma.delta=runif(1), sigma.gamma=runif(1), sigma.y=runif(1))
}
parameters <- c("gamma", "delta")

pilots <- jags(data, inits, parameters, pilot_model, n.chains=3, n.iter=10000, DIC = F)

## extract sd and 50% intervals, 95% intervals
delta_sd <- pilots$BUGSoutput$sims.matrix[, 1:8] %>% apply(., 1, sd)
sigma_sd <- pilots$BUGSoutput$sims.matrix[, 9:13] %>% apply(., 1, sd)

mean(delta_sd)
quantile(delta_sd, probs = c(.025, .25, .75, .975))

mean(sigma_sd)
quantile(sigma_sd, probs = c(.025, .25, .75, .975))
```

```
> mean(delta_sd)
[1] 0.3209558
> quantile(delta_sd, probs = c(.025, .25, .75, .975))
     2.5%       25%       75%     97.5% 
0.2398945 0.2945896 0.3485017 0.3971116 
```

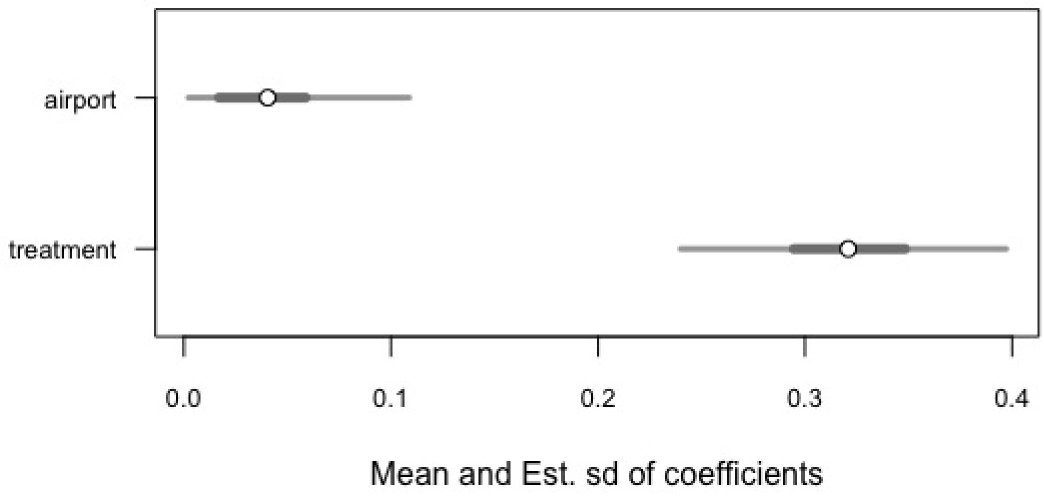
```
> mean(sigma_sd)
[1] 0.04040801
> quantile(sigma_sd, probs = c(.025, .25, .75, .975))
       2.5%         25%         75%       97.5% 
0.002125195 0.016783150 0.058750085 0.108773172 
```

```
# plot
quant <- c(.025, .25, .75, .975)
data_plot <- matrix(c(mean(delta_sd), quantile(delta_sd, quant),
                   mean(sigma_sd), quantile(sigma_sd, quant)), nrow = 2, byrow = T) %>%
 data.frame
colnames(data_plot) <- c('mean', '2.5%', '25%', '75%', '97.5%')
rownames(data_plot) <- c('treatment', 'airport')
data_plot$y_pos <- c(2, 4)

plot(range(data_plot[, 2:5]), c(1, 5), type='n',axes=F,ylab='', xlab='Mean and Est. sd o
f coefficients')
axis(1,cex.axis=.8)
axis(2, at = c(2, 4), labels = rownames(data_plot), cex.axis=.8, las=2)

apply(data_plot, 1, function(x) segments(x[2], x[6], x[5], x[6], lwd=3, col='grey65'))
apply(data_plot, 1, function(x) segments(x[3], x[6], x[4], x[6], lwd=5, col='grey50'))

apply(data_plot, 1, function(x) points(x[1], x[6], pch=16, col='white', cex=1))
apply(data_plot, 1, function(x) points(x[1], x[6], pch=1, cex=1.1))
box()
par(lend=0)
```



The dataset of Pilot.data was used for this question. This is a varying-intercept multilevel dataset with two group-level indicators: airport and treatment. After fitting the model using JAGS, an ANOVA plot showing the estimated finite population standard deviations with 50% and 95% uncertainty intervals was plotted. It can be noted that treatment is more important in variance than airport.